NSI	Représentation des données : types et valeurs de base.	TP7

1. Écriture d'un entier positif en base b :

Les entiers en Python ne sont pas des entiers "machine" sur 32 ou 64 bits, ce sont des entiers mathématiques "non bornés". On peut cependant demander d'afficher un entier en binaire ou en hexadécimal, et saisir un entier en binaire ou en hexadécimal.

Exercice 1 : Ouvrir l'interpréteur interactif IDLE, saisir les instructions suivantes et recopier le résultat.

instruction	Résultat
`hex(42)`	
`bin(42)`	
`0x11`	
`0b11`	

2. Valeurs booléennes et opérateurs booléens :

On peut signaler que Python, sous l'influence du langage C, convertit à la volée les entiers en booléens et vice-versa

Exercice 2 : Toujours dans l'interpréteur interactif IDLE, saisir les instructions suivantes et recopier le résultat dans la colonne résultat.

instruction	Résultat
`0 == False`	
`1 == True`	
`1 + True`	
`if 1 : print('1->True')`	
`x = 0` `if x != 0 and 1 / x == 3 : print('x vaut 1/3')`	
`if 1 / x == 3 and x != 0 : print('x vaut 1/3')`	

3. Représentation d'un texte en machine :

On peut retrouver le code unicode d'un caractère ou obtenir le caractère correspondant à un code donné avec les fonctions `chr` et `ord`

Exercice 3 : Toujours dans l'interpréteur interactif IDLE, saisir les instructions suivantes et recopier le résultat dans la colonne résultat.

instruction	Résultat

`chr(32)`
`chr(64)`
`chr(257)`
`chr(49829)`	

Exercice 4 : En vous inspirant de l'**exercice 3** faites afficher le code ASCII des caractères **'A'**, **'D'**, **'0'** (zéro), **'3'** mais en utilisant cette fois-ci l'instruction **ord(' ')**, puis vérifiez sur la table ASCII ci-dessous; voyez-vous comment on peut déduire le code ASCII de toute lettre majuscule à partir de celui de **'A'**, et de tout chiffre à partir de celui de '**0**' ?

Code ASCII

32	33 !	34 "	35 #	36 $	37 %	38 &	39 '
40 (	41)	42 *	43 +	44 ,	45 -	46 .	47 /
48 0	49 1	50 2	51 3	52 4	53 5	54 6	55 7
56 8	57 9	58 :	59 ;	60 <	61 =	62 >	63 ?
64 @	65 A	66 B	67 C	68 D	69 E	70 F	71 G
72 H	73 I	74 J	75 K	76 L	77 M	78 N	79 O
80 P	81 Q	82 R	83 S	84 T	85 U	86 V	87 W
88 X	89 Y	90 Z	91 [	92 \	93]	94 ^	95 _
96 `	97 a	98 b	99 c	100 d	101 e	102 f	103 g
104 h	105 i	106 j	107 k	108 l	109 m	110 n	111 o
112 p	113 q	114 r	115 s	116 t	117 u	118 v	119 w
120 x	121 y	122 z	123 {	124 |	125 }	126 ~	

Exercice 5 : En vous inspirant du code à compléter ci-dessous, faites calculer à Python la position dans l'alphabet de la lettre M (réponse : 12 si on compte à partir de 0, sinon 13 pour les humains) ;

```
def ieme_lettre(i):
    return …

print(ieme_lettre('M'))
```

Exercice 6 : Faites afficher à Python les caractères de code ASCII compris entre 32 et 44 sur une même ligne

(Réponse : !"#$%&'()*+,).

Exercice 7 : Modifier le code de la question précédente de manière à ce que l'affichage se fasse de manière verticale (voir ci-contre).

⇨

!
"
#
etc…

Exercice 8 : Définissez une fonction **est_chiffre(c)** qui renvoie **True** si le caractère **c** est un chiffre. Vous n'utiliserez pas la méthode isdigit, ce serait de la triche. Bon, ou alors uniquement une fois que vous avez vu comment faire sans.

Exercice 9 : Définissez une fonction `masque_numero(s)` qui renvoie la chaîne s dans laquelle les chiffres sont remplacés par des étoiles.

```
masque_numero('Bonjour je vends 1 chat. Appellez au 0678912345.')
'Bonjour je vends * chat. Appellez au ********.'.
```